BOOTABLE NAND FLASH MEMORY ARCHITECTURE

BACKGROUND OF THE INVENTION

The subject invention is directed broadly to memory architectures, and more

5 particularly to memory architectures in high-density or low-power applications, and will be

described with particular reference thereto. It will be appreciated by one of ordinary skill in

the art that the subject architecture is useful in any fetch-instruction processor application in

which lower cost or higher density is advantageous.

Typical processor architectures employ a central processing unit ("CPU") working in

10 conjunction with both addressable and non-addressable memory. A typical, addressable

memory includes two sets of lines to access specified memory locations. Address lines allow

for input of a binary number that corresponds to a memory location. In an addressable

memory, a particular memory location is specified by placing a binary address as a digital

input on the address lines. At this point, access by the data lines is made to the specified

15 memory location. A read-write enable ("RW") input is typically used to signify whether data

from the data lines is to be stored into the selected memory location, or read from the selected

memory location. Since any address may be placed on the address lines at any given time,

access is truly random in nature. Thus, an addressable memory devices is referred to as

random access memory, or RAM. A non-addressable memory by comparison does not use

20 address lines to select a particular memory location. Instead, data is written into the memory

or read from the memory as a serial stream of information. An early tape drive for storage of

data would be one example of a non-addressable memory device.

Low-power or compact CPUs are typically formed as microprocessor units ("MPU").

Data, which is stored in the memory, is communicated in parallel by a plurality of data lines.

25 MPU is used herein insofar as a microprocessor unit as envisioned in the preferred

embodiments. However, it will be appreciated by one of ordinary skill in the art that the

subject architecture is applicable to any digital CPU/memory architecture.

During typical processor operation, a CPU provides an address in the memory device

from which a particular data item or instruction is desired. The memory device retrieves the

30 data or instruction from the selected location via the address lines, and provides that data or

instruction on the data lines. Thus, the MPU is able to fetch the desired data or instruction from the device. Such an addressable memory architecture works in a random fashion, leading to its designation as random access memory RAM, as noted above.

Other conventional memory devices are not random access in nature. Such data devices may require serial access of data, such as from a tape drive, hard disk, serial data stream or the like. Given the serial nature of such data streams, it is difficult to use a sequential memory device for fetching and processing instructions or data.

Most digital processing systems are prepared to become active from an off or powered-down state. When powering up, a digital device has extremely limited capability, and must at least load a core or kernel of instructions to allow the device to engraft upon the system the functionality necessary to accomplish most conventional, computerized tasks. This is generally accomplished by having a forced memory location from which a processor will commence operation during a power-up sequence. This memory location has a set of instructions which provides the basic functionality necessary for the system. Once this initial application has been loaded and run, the system has enough intelligence to allow it to accomplish more refined tasks, such as addressing peripheral equipment, loading an operating system or commencing operation of other, selected programs.

The operation of a typical boot sequence, as noted above, requires a presence of at least a small core of RAM to allow for sequential addressing of a boot code. Random access memory is relatively expensive and requires a relatively large amount of space to fabricate when compared to serial memory devices. Such a serial memory device, e.g., NAND flash memory, is still used in hybrid RAM/NAND architecture. It would be desirable if a boot sequence could be completed from NAND memory, thus eliminating the requirement of RAM memory with its associated drawbacks.

The subject invention provides a bootable CPU that is able to use code disposed in NAND flash memory. Thus, power applications such as personal digital assistance ("PDA"), cellular phones or other compact and energy efficient electronic devices may utilize NAND flash memory with its associated benefits.

An advantage of the present invention is the provision of a processing system which utilizes less power than those employing RAM memory.

Another advantage of the present invention is the provision of a processor system that is more compact.

Yet another advantage of the present invention is the provision of compact and cooler-memory which allows for selective inclusion on the same chip substrate of that of a processing
5 device.

These and further advantages will become apparent to others upon reading and understanding the subject disclosure.

BRIEF SUMMARY OF THE INVENTION

10 In accordance with the present invention, there is provided a non-addressable memory for outputting a code sequence which includes a series of data storage locations. A set of data lines is adapted to output data stored in the data locations. A device includes an internal address register which stores an address associated with a memory location from which data is output on the output lines. A boot sequence is commenced by placing a preselected address
15 into this register upon receipt of a run signal. This run signal is representative of the commencement of a preselected code sequence, e.g., the boot sequence. A counter is provided for selectively incrementing the internal address register such that a series of instructions may be output on the data lines. When this sequence of instructions correlates to timing of instructions as expected by an associated CPU, the CPU can commence operation of a
20 sequence, such as a boot sequence, without the use of a memory inclusive of address lines.

In accordance with another aspect of the present invention, a mechanism is provided for synchronizing the CPU instruction fetch with an output of a next instruction from the sequence of data storage locations.

In accordance with the further aspect of the present invention, the loading of the
25 preselected address into the material address register is commenced with a power-on signal associated with a boot of a data processing device.

BRIEF SUMMARY OF THE DRAWINGS

Turning now to the drawing wherein the depictions are for the purpose of illustrating the preferred embodiment of the invention, and not for the purpose of limiting the same:
30

Fig. 1 illustrates a physical connection between a microprocessor unit and the non-addressable memory as taught by the subject invention;

Fig. 2 illustrates an embodiment as with Fig. 1, but inclusive of a second chip enable line to allow for communication between the MPU and serial memory device;

Fig. 3 illustrates additional signal lines disposed between essential processor unit and a serial memory device;

Fig. 4 illustrates a load sequence to provide for initial register loading within a serial memory device as provided at boot time; and

Fig. 5 illustrates a representative timing diagram for address and communication lines provided between the central process unit and serial memory device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Turning now to the drawings, Fig. 1 illustrates the overall architecture of the embodiment of the subject invention. A processor device A includes a CPU which, in this embodiment, is comprised of the MPU 10. In the preferred embodiment, an MPU is comprised of a sequential processor consisting of a reduced instruction set computer, or RISC device. As will be appreciated by one of ordinary skill in the art, a RISC processor operates with a smaller instruction set than that employed by a complex instruction set computer ("CISC"). Virtually every logical operations of a CISC processor is realizable by a RISC processor operating with a greater number of discrete operations. Given the reduced hardware required by the reduced instruction set, a RISC processor operates more quickly, usually far in excess of 500 MHz. Relative high frequency of a RISC device requires a fast, sequentially accessible memory to provide to it a rapid stream of instructions. Absent such a rapid stream, wait states must be injected into the MPU to allow for a valid, next instruction to be present before execution thereof.

The MPU 10 is in data communication with a non-addressable memory device, suitably comprised of a NAND memory 12. A NAND flash memory, named after the negated-AND gate that forms a fundamental component of the logic of its construction, is a relatively low cost, high capacity, non-volatile memory device ("NVM"). A NAND flash memory is typically greater than one Gigabit of storage capacity. Give that a NAND flash

memory is non-addressable, it is typically thought of for use as data only storage device. An alternative memory structure, known as NOR flash is named from its use of a negated OR gate forming its fundamental component, is traditionally thought of as an addressable memory device for use in instruction storage applications, or hybrid instruction and data storage applications, due to its random access storage capabilities.

In view of the forgoing, many systems employed NOR flash devices to store exectuable code, such as boot code, and NAND flash for data storage, particularly bulk data storage, such as may be needed to store data files, such as spreadsheet information, word processing files, database files, as well as multimedia files, such as image files, audio files, video files, and the like. By way of example, such systems include: personal digital assistants ("PDAs"); cellular telephones, particularly multimedia phones; set-top boxes; digital cameras; and digital recorders, as well as any analogous digital device.

The subject invention allows for a NAND memory device to accommodate both the storage of data, as well as instructions. The subject system thus allows for a single memory device to be used in applications as is noted above. This is particularly advantageous given that such applications may only require an extremely limited amount of code memory, especially in relatively small boot code requirements, while many manufactures no longer fabricate low capacity NOR flash. This further results in a waste of space and expense for unused NOR flash memory capacity.

A plurality of data lines 14 provide for interchange of data between NAND memory 12 and MPU 10. In the illustrated embodiment, sixteen (16) data lines are provided D0-D15, which allows for transmission of a byte or word of sixteen (16) bits, in parallel. It will be appreciated by the illustration of Fig. 1, the non-addressable memory device, as illustrated with the NAND flash device, does not have any dedicated address lines 16 for direct connection to the MPU 10. In such an architecture, there is no mechanism in a conventional sense to allow for a NAND memory 12 to indicate to the MPU 10 that it has a relatively long access latency. In a typical, conventional NAND flash memory, such latency may be around 25 microseconds.

In accordance with the subject invention, the memory device such as NAND memory 12 is provided with a sequence mode in addition to normal operation mode. This sequence

mode facilitates output of sequential memory data to the system when the host system, such as MPU 10, requests data by asserting address on its output enable pin OE#, or a similar pin such as read enable pin RE#. Thus, a memory device such as NAND 12 may commence a selected code sequence output upon receipt of a commencement signal.

5   The sequence noted above is suitably activated by a dedicated chip enable to signal CE2# which is suitably assertive on a memory device or power-on reset detection circuitry by a reset pin, such as that illustrated by RST#. Such signals suitably generates an internal signal within the memory device 12 to enable the afore-described sequence mode. As will be apparent from a review of Fig. 1, the address lines 16 are not used in a conventional sense as

10   would be with an addressing scheme in a random access memory device.

Turning now Fig. 2, a variation of the architecture described in Fig. 1 is provided. In Fig. 2, a processing device B includes a central processor unit, such as MPU 20, in data communication with a non-addressable memory, such as NAND memory 22. In a particular embodiment of Fig. 2, the memory 22 includes first and second chip enable input CE1# and

15   CE2#. These lines correspond to outputs from the processor 20 formed of first and second chips select lines, CS# and CSn#, respectively. An output enable OE# from the MPU 20 forms a read enable RE# into memory 22. In this embodiment in which a second chip enable signal line is provided, a mechanism is provided when a microprocessor may start an initial boot up and the memory device 22 is suitably set to a sequence mode, as described above in

20   connection with Fig. 1, by assertion of CE2#, and then returned to a normal mode of operation by a negation of a signal on CE2#. When the memory device is provided with power-on or reset detection circuitry, it will be appreciated by one of ordinary skill in the art, after a power-on CE# of the memory device is ignored and the memory device is activated to operate in the afore-noted sequence mode without regard to the CE# input. In such an operation, a sequence

25   to return the memory device to normal mode is necessary. A suitable method accomplishing this is to send a command to the memory device to terminate the sequence mode, as illustrated in Fig. 2.

Turning now to Fig. 3, an alternative embodiment for commencing a sequence mode is illustrated. As provided in Fig. 3, an MPU 30 and non-addressable memory 32, such as

30   NAND flash memory, are illustrated. It will be appreciated that the additional

interconnections, such as that provided by data lines, have been omitted from Fig. 3 for ease and illustration. In the embodiment of Fig. 3, a reset input into memory 32 is illustrated by RSTIN#. A reset output from memory 32 into processor 30 is illustrated by RSTOUT#. Such additional communication between the processor and memory device allows for an alternative action of a sequence mode, such as a boot sequence. This may be accomplished by providing a commencement of a boot signal by placement of the RSTIN# signal, the commencement of which generates the RSTOUT# signal into processor 30. Thus, the reset signal commences the operation, such as the boot sequence operation, and is synchronized with operation of the processor 30.

In accordance with the foregoing, it will be appreciated that an output signal from a memory device is suitably used to indicate that the device is busy and may be used to prevent an associated processor from timing out. By way of example, a microprocessor is suitably kept in reset mode until boot strap code is ready in the associated memory device. For situations, such as a soft reset, suitable commands are provided to a NAND flash to address situations in which a re-boot may be necessary without power cycling.

The sequence mode noted above includes two functions. A first function is to enable serial data output from a memory device. This is suitably accomplished by toggling the output enable OE#, or a similar signal, such as the read enable RE#, of an associated microprocessor. In the preferred embodiment, the data represents sequential microprocessor instructions and thus no address pins are required. When a sequence is enabled, the memory device in the preferred embodiment automatically loads its internal register from an associated memory cell area which holds initial boot strap code as is illustrated in Fig. 4. As illustrated therein, a memory cell 40 includes a boot strap program area 42 which is selectively placed in data communication with a register 44. A memory cell area 40 is suitably a primary memory cell or alternatively isolated from a primary cell memory area, such as a redundant memory cell array. The register area 42 provides for addressing within the memory cell 40. It is to be appreciated that even the designated internal register 44 may be used to store data. When used in a boot sequence, a boot strap code size may vary from one page length, in the case of NAND flash memory to more than one page length by use of sequential read functions, or other analogous functions such as gapless sequential read functions typically provided with

NAND flash memory devices as will be appreciated by one of ordinary skill in the art. It will therefore be appreciated that an initial boot strap sequence can be any suitable length as may be required for a particular application.

It will also be appreciated that the memory device may be provided with other adjuncts as is typically provided, such as error correction code, parity bits, as well as any ward length which may be desirable for a particular application.

In connection with the foregoing, it will be appreciated that a selected sequence, such as a boot program, is advantageously written without any jump instructions insofar as there is no mechanism to communicate address output from a microprocessor. The system functions to output data sequentially, assuming that a sequence or a boot program execution address does not jump out of sequential order. Practically, this is not a problem, insofar as one purpose of a program, such as a boot program, is to set internal registers of microprocessor and/or associated system. In typical operation, such actions do not require any loop, jump or conditional jump instructions.

In accordance with the subject system, in the event that code sequence are forced to utilize loops, jumps, conditional jumps, or the like, a boot program may suitably copy itself into associated RAM. Such a copy function is provided itself from sequentially executing instructions. After completion of such a copying operation, the boot program may transfer execution to that portion of the program which has been copied into an associated RAM.

Turning now to Fig. 5, a suitable timing diagram for address lines, chips lines, output enable lines and data lines, as illustrated in the foregoing figures, is described. With the relative timing as illustrated in Fig. 5, it would be appreciated that memory such as NAND flash devices which do not possess address pins may be directly tied to a processor bus which has typical bus timing as illustrated by the figure. Thus, the system provides for use of boot memory without address pins and is particularly advantageous for allowing for a boot sequence to be completed from a NAND memory device.

By virtue of the foregoing, a mechanism is provided for allowing for power-up, reset or other specialized instruction sequences to be accomplished directly from a serially or sequentially outputting memory, such as a NAND flash memory, as well as any other memory that does not provide for direct addressing via address lines. Thus, devices are fabricated

which may be lower cost, smaller dimension, generate less heat, and be less expensive to operate than have been provided heretofore.